

Management of multimedia resources: from a generic information model to its application to an MPEG2 video codec

Constant GBAGUIDI, Simon ZNATY, Jean-Pierre HUBAUX and Olivier VERSCHEURE

Swiss Federal Institute of Technology

TCOM Laboratory/ Telecommunications Services Group

CH- 1015 Lausanne, Switzerland

Phone : +41 21 693 56 13 / Fax : +41 21 693 26 83

E-mail : {gbaguidi, znaty, hubaux, verscheure}@tcom.epfl.ch

Abstract

New open service architectures provide a management framework for telecommunications services, telecommunications networks and computing resources. However, the introduction of multimedia applications in these architectures will require the management of the underlying multimedia resources (e.g., codecs, converters, etc). Multimedia resources are the basic components that support multimedia communications. In this paper, we tackle this issue by proposing a generic management information model for multimedia resources and then instantiate it for the management of an MPEG2 video codec. This information model provides a data representation of the multimedia resources in order to manage them efficiently.

Keywords

Resource management, information model, MIB, SNMP, MPEG 2, video codec.

1 INTRODUCTION

The prodigious growth of the computer market these last years has entailed the development of novel applications, especially multimedia applications whose particularity is to combine several information types such as speech, video, graphics and text. In order to fully exploit this opportunity, computer vendors, yet offering multimedia-ready platforms and upgrade kits within stand-alone desktops, are currently providing portable computers with the same sophisticated tools. Therefore multimedia is becoming part of daily work schedules, and increasing numbers of computer users are getting familiar with multimedia computing. As a

consequence, the needs of connectivity with other multimedia users and sources across local and wide area networks are increasing. Therefore the new trend is towards networked multimedia applications. These applications are of real time and multipoint nature, and require large amount of processing, communication and storage resources.

In this context, several multimedia resources have to be provided, such as video and audio codecs, media converters (e.g., conversion from text to audio), echo cancellors, and so forth. Since all resources (network resources as well as multimedia ones) involved in a communication session take part in the provision of the required Quality of Service (QoS), the characterization of multimedia resources becomes a necessary issue to be studied in order to identify the accurate parameters which are relevant for managing those resources' activities. Multimedia resources (also called information converters or special resources in some literature) are different from network resources, in that their processing tasks can highly affect the content of the incoming information. These observations result in the necessity for providing multimedia resources with a management framework. It is worth noting that this issue has not yet been tackled by the management community. To date, emphasis has been put on service management, network management and computing resources management, and on their integration. However, managing multimedia services will definitely require the management of multimedia resources.

We have investigated *two approaches* for describing the management information related to multimedia resources. *First*, we have proposed a multimedia resource Management Information Base (MIB) (Gbaguidi, 1995), which contains the relevant objects for the management of multimedia resources. Our MIB has been designed to be compatible with the SNMP management protocol and thus has been specified using SMI (RFC1155, 1990). *Secondly*, we are proposing in this paper an object oriented specification of a generic information model of multimedia resources, i.e, the representation of multimedia resources in terms of data much easier to read and providing a more structured view of the information to be managed.

This paper is organized as follows. In Section 2, the resource management scheme is presented, in order to get a fix on the management paradigm and the used protocols. In Section 3, a generic information model of multimedia resources is proposed. Finally, in Section 4, the designed model is specialized for a specific multimedia resource, namely an MPEG2 (Moving Pictures Experts Group) (ISO-13818-2, 1994) video codec.

2 THE MANAGEMENT MODEL OF RESOURCES

The rationale behind using multimedia resources is the provision of some QoS. Therefore it is worthy to study how users requests are issued to these resources in some meaningful terms (Figure 1). The management invocations from the *high level client* are issued to the resource via the *resource manager* using CMIP, (ITU-T X.711, 1991) and (Voruganti, 1994). To perform the requested operation, the *resource manager* needs access to the *managed resource*. This latter is modelled by an *agent* and a *MIB*. The *agent* is the only entity allowed to access the *MIB*. The protocol generally used to invoke the managed object is SNMP, (RFC1157, 1990) and (Voruganti, 1994), thanks to the latter's simplicity. Indeed SNMP is a polling oriented protocol, very easy to deploy, in contrast with CMIP whose wide range of functionalities (security, fault management and so forth) undermines the implementation by equipment vendors.

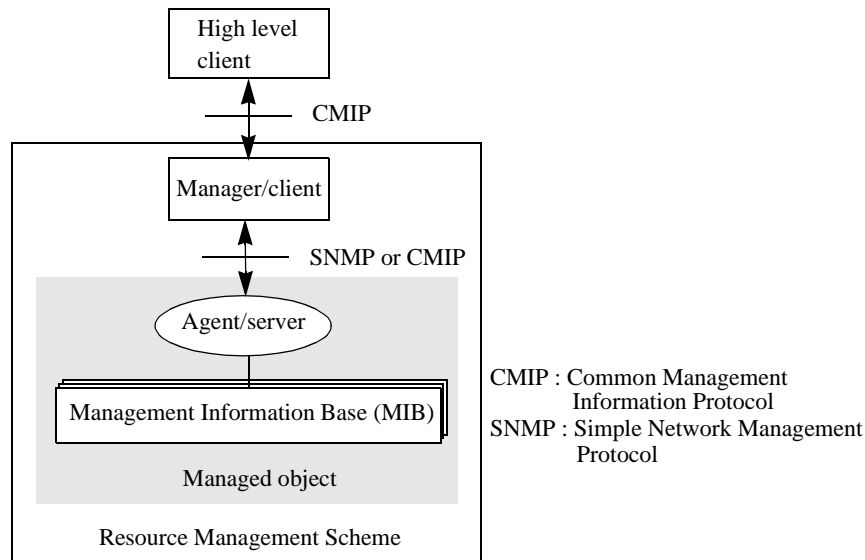


Figure 1 Illustration of the OSI resource management scheme.

In the management model, the role of the resource agent is that of a supervisor. The same agent can monitor several MIBs. On the contrary, the manager's role is more complex. To illustrate this, we have modified the *L-E (Legislator-Executor) model* borrowed from (Chan, et al., 1995), by including a *supervisor* entity (Figure 2). The rationale for choosing the L-E model is that, in spite of its simplicity, it highlights the major functionalities any resource control system should carry out. However, the entire management framework proposed in relation with the L-E model (Chan, et al., 1995) does not strictly match the OSI management scheme. Indeed, the framework proposed in (Chan, et al., 1995) does not feature a resource agent nor a MIB. To overcome this lack we are proposing in Figure 2 an enhanced version that matches more closely with the OSI management scheme. Interfaces are clearly defined and complex processes are accomplished by the manager.

The proposed L-E model highlights a data store of *management parameters* which manages the overall management task. These parameters express constraints such as the QoS expected from the resource in a high level "language". Therefore, the resource manager must perform the mapping of the management parameters into *control parameters*, which are more meaningful to the resource. This mapping functionality is achieved by the *parameter translator*. The resulting control parameters influence the remainder of the model, as they constitute additional constraints over the provision of the resource service. The other basic constraints are the *resource capacity* (computed by an entity called *capacity estimator*, e.g., in the shape of a multimedia capacity region (Lazar, et al, 1995)), and the traffic (*request intensities*) at the resource access points. The *legislator* is responsible for stating access rules (*control policy*) on the resource after considering the entire set of constraints. The stated control policy is used by the *executor* to regulate the resource activities. All the mentioned components are located inside the *resource manager*. A huge amount of the intelligence provided by the manager is achieved by the parameter translator and the legislator entity. Indeed, the mapping of management parameters onto control parameters may induce trade-offs between some of the resource features. For instance the realization of high compression ratios sometimes requires long processing delays, making the trade-off between the bandwidth and the processing delay difficult to optimize. As such, the parameter translator is an intelligent agent, i.e., an expert system which may use the previous behaviour of the resource for making accurate trade-offs. Likewise, the legislator may also use its training or experience for setting the appropriate control policy.

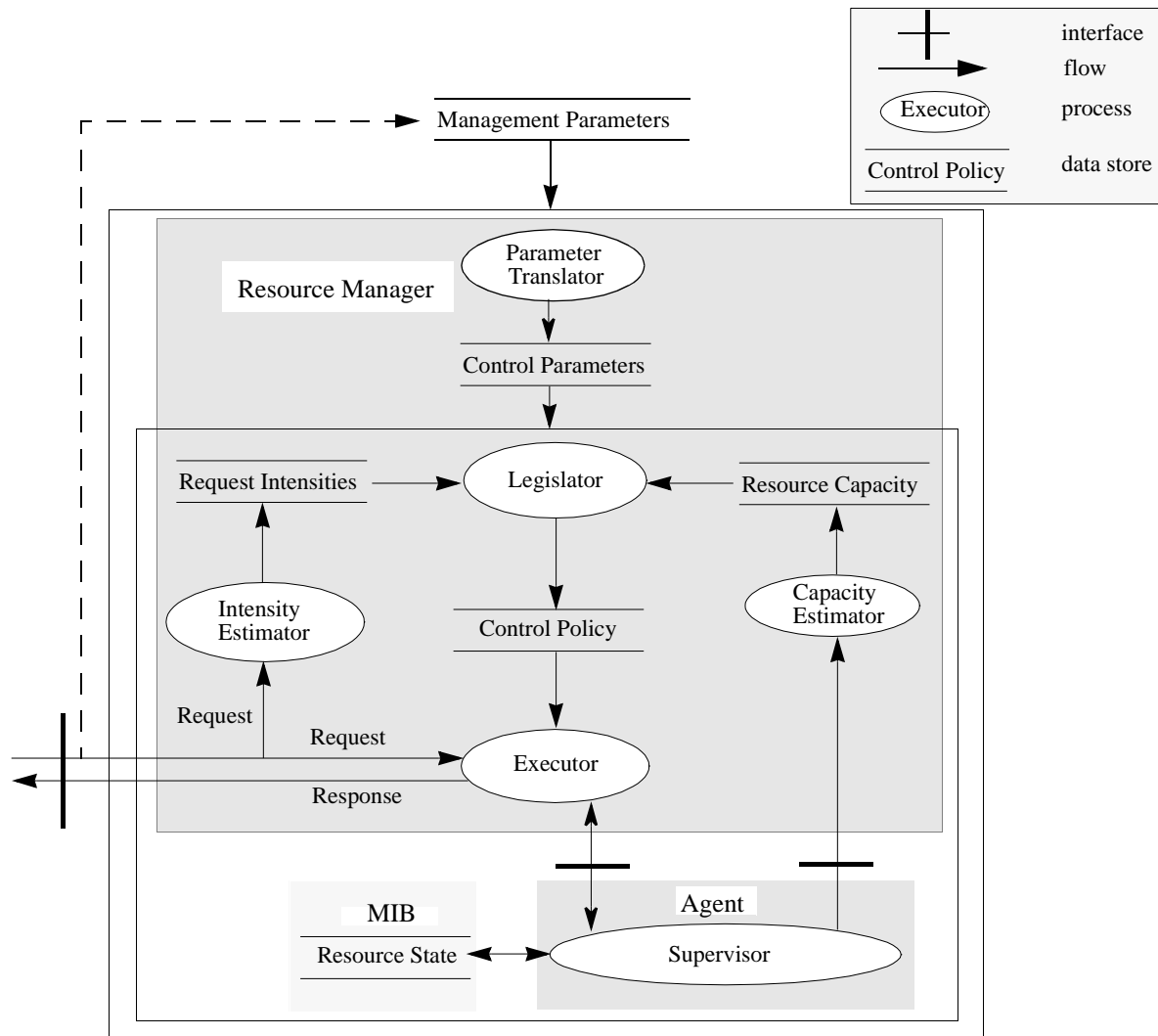


Figure 2 A modified L-E model matching the OSI management scheme. The original L-E model can be found in (Chan, et al., 1995), CTR, Columbia University, NY.

The *agent* is represented by the *supervisor* process. It polls the *resource state* and provides information to the *executor* and the *capacity estimator*. The advantage for using the L-E model is mainly its simplicity. Complex issues such as security and accounting are declined as management parameters, thus keeping the model very simple. The resource manager can then be modelled as in Figure 3.

Every resource manager is composed of two generic elements: the *service* supporting the provision of the management requests from high level clients, and the *software* in charge of coordinating the manager's activities. The service object consists of a set of *functions*. Among the basic functions provided by the manager are *capacity estimation*, *traffic estimation* and *management parameters translation*. Each function is characterized by a set of *abstract parameter schemes*. For instance the traffic estimation function depends on the number of requests coming through the resource accesses, and so forth. Parameters are generally expressed by means of schemes in the case of multimedia resources. In particular the result from capacity estimation is often a scheme (e.g., a multimedia capacity region (Lazar, et al, 1995)) taking into account several parameters.

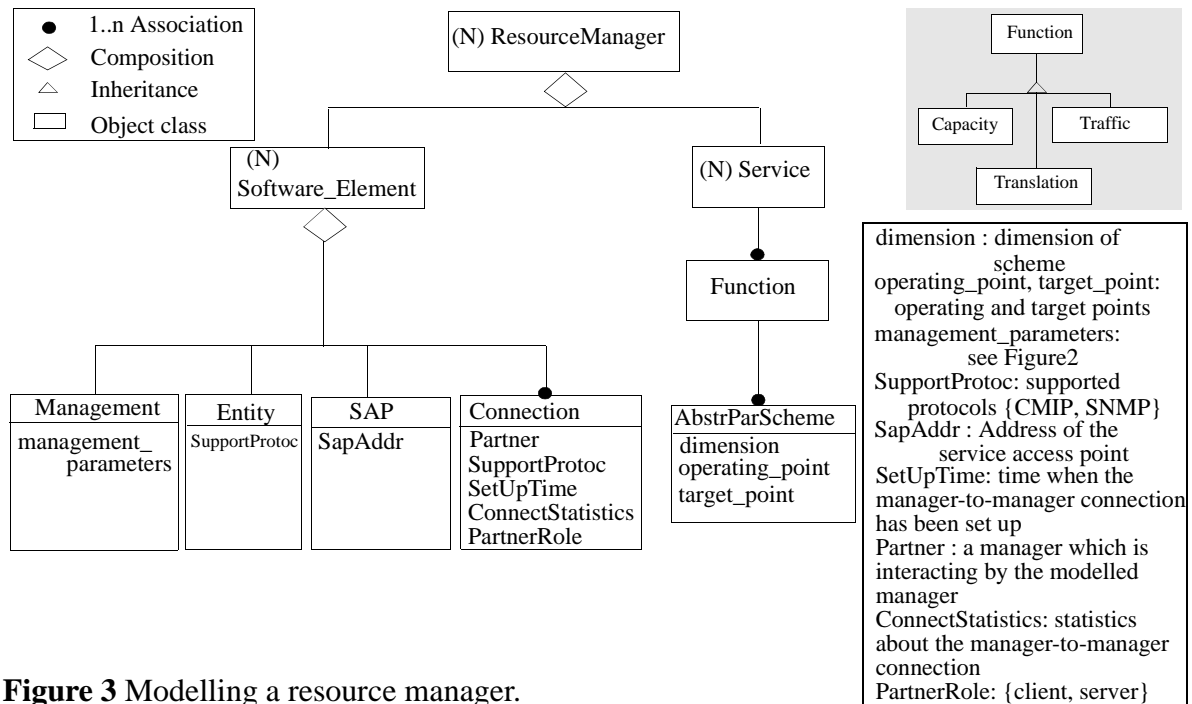


Figure 3 Modelling a resource manager.

The software element of a resource manager aggregates four classes, namely *management*, *entity*, *SAP* (Service Access Point) and *Connection*. The former is concerned with the management parameters as introduced above. It enables the management of the manager. The *entity* class is mainly concerned with the behaviour of the manager in terms of the parameters needed for computing statistics about the manager. The *entity* class relies on the protocol used to issue requests to the manager via its *SAP* instance. Finally the *Connection* class describes a manager-to-manager interaction, i.e., the interaction between the manager of interest and another one.

Functions provided by the manager together enable the granting of the high level client requests. To do so they need information from the MIB. The ensuing Section outlines the useful generic information needed for serving the high level clients in a satisfactory way.

3 A GENERIC INFORMATION MODEL OF MULTIMEDIA RESOURCES

As characterized in (Sclavos, et al., 1994), an information model is intended to have some properties, especially information structuring, abstractness, genericity and implementation independence. Structuring the information enables to highlight all the components of the modelled element. Abstraction and genericity emphasize the commonalities among entities, such that the model can be specialized or instantiated for many resources by means of object specialization or inheritance.

In addition, any information model must support the basic management functions identified by ISO, namely configuration, fault, security, performance and accounting. In (Sclavos, et al., 1994), an abstract information model of network elements has been designed that complies with the mentioned properties. Therefore, our approach has been to derive from that work the most suitable information model of multimedia resources, after outlining the main characteris-

tics of these resources and the differences existing between them and the network resources.

3.1 Characterization of multimedia resources

Multimedia resources can be realized in software, hardware or in both software and hardware. In (Rodriguez, et al., 1994), it has been stated that software-only video codecs, yet less complex than hardware-assisted codecs, are more limited in quality. Thus the best compromise seems to be an hybrid design. Anyway, in order to provide for management schemes, resources have to present a clearly defined interface with the supported protocols. Although the most relevant information one essentially needs from a resource is the activities at its accesses, in some cases it might be interesting to know of the resource internal activities. Therefore *two concepts* need to be introduced. First, the concept of *resource activity modes* describes the external view of the resource's activities. By contrast, the second concept of *resource activity layers* refers to the internal activities leading to the provision of the overall resource's functionality. For instance, certain audio CD players use one-bit sampling. The user's view of these devices is the playing quality, whereas an internal view shows the processing steps, namely the oversampling (one-bit sampling) and the buffering of the resulting bits to constitute an audio data unit.

As far as resource management is concerned, resource activity modes are the main features. However, as mentioned above, the device technology can influence the quality of the provided service. For instance, a one-bit sampling device requires less electronic components than does a full-bit sampling device, thus being easier to manage. Therefore the resource complexity level can be a selection criterion.

As a consequence of considering resource activity layers, some multimedia resources can be aggregation of some others. Thus, the *recursiveness* property, well known in the network management field, can be applied, which states that a multimedia resource may contain many others. In the CD player example, the buffer used to store the information bit by bit can also be considered as a multimedia resource.

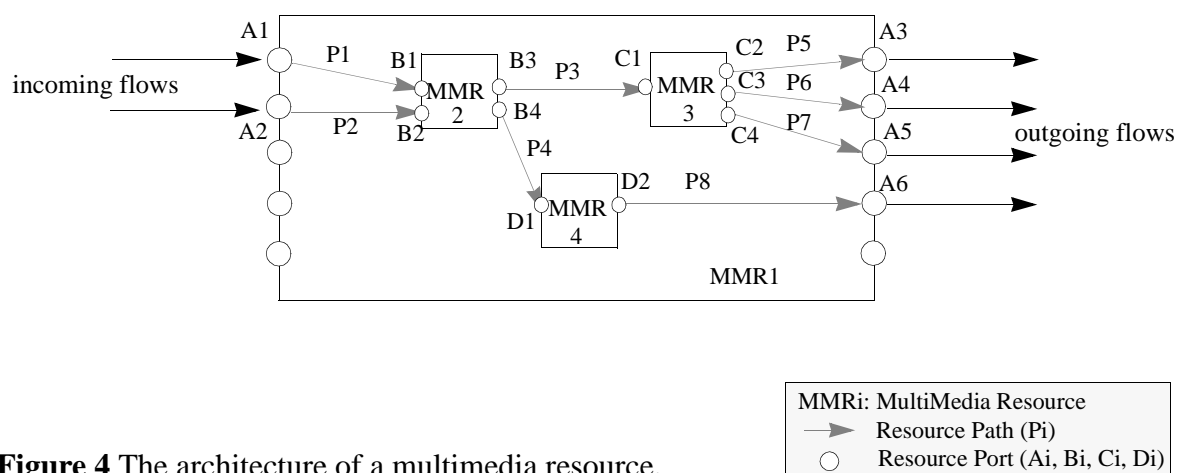


Figure 4 The architecture of a multimedia resource.

Furthermore, multimedia resources aggregate ports, paths and processing elements (Figure 4). These usually need to know the real nature of the flow that is about to be processed. For instance MPEG codecs do not process both audio and video information in the same way. This makes a big difference with the classical network elements. This difference is essentially due

to the fact that multimedia resources are service resources, so that they do not pertain to the same abstraction (or management) level together with network resources. While protocols are important at the network level, the service level is mainly concerned with the “native” information’s features.

Multimedia resources, especially those assisted by some hardware, are characterized by their distortion-safe operating domain, their technology, their processing time, their resolution, their reliability, their availability, their transient behaviour, their lifetime, their evolution schemes, their compatibility with existing standards, and so forth. These features influence the choice and the performance of multimedia resources handling the required QoS.

The above characteristics are abstract parameters to refine for each specific resource. All of them are not needed to describe a given function. For instance, performance evaluation does not depend on the technology in support of the device, and so on.

3.2 The proposed generic information model

The designed model (Figure 5) illustrates the discussion of the previous subsection. Activity modes appear in terms of flows (*InputFlow* and *OutputFlow*), whilst activity layers are mainly represented by low level multimedia resources, called (*N-1*) *MultimediaResources*, assuming that the level of the container resource is *N*. Activity layers can also be expressed in terms of abstract parameters. In the following, each aggregated entity is described.

The Architecture Element class

As stated above, multimedia resources may be designed in software and/or hardware. Therefore their architecture may consist of a software part and/or a hardware part. This latter is an equipment characterized by its description, its physical location, the contact information in case of problems, and the name under which the device is administered.

Concerning the software element, it aggregates four classes, namely *management*, *entity*, *SAP* (Service Access Point) and *Connection*. The *Management* class covers the information needed to manage the resource. The *Entity* class is concerned with the behaviour of the resource. It includes the information needed to compute statistics about the resource. *SAP* represents the point from which the service provided by the resource can be requested. The *SAP* relates to the protocol used or a functional addressing plan. Finally the *connection* class illustrates the connections among the entities aggregated by the modelled multimedia resource. An example set of *connection* objects is provided in Table 1, which corresponds to the multimedia resource architecture depicted in Figure 4.

Table 1 Example Connection Table of Multimedia Resources

Input MMR	Source Port	Output MMR	Sink Port	Path Identifier
MMR1	A1	MMR2	B1	P1
MMR1	A2	MMR2	B2	P2
MMR2	B3	MMR3	C1	P3
MMR2	B4	MMR4	D1	P4
MMR3	C2	MMR1	A3	P5
MMR3	C3	MMR1	A4	P6
MMR3	C4	MMR1	A5	P7
MMR4	D2	MMR1	A6	P8

In the above table, each row represents a *connection* object.

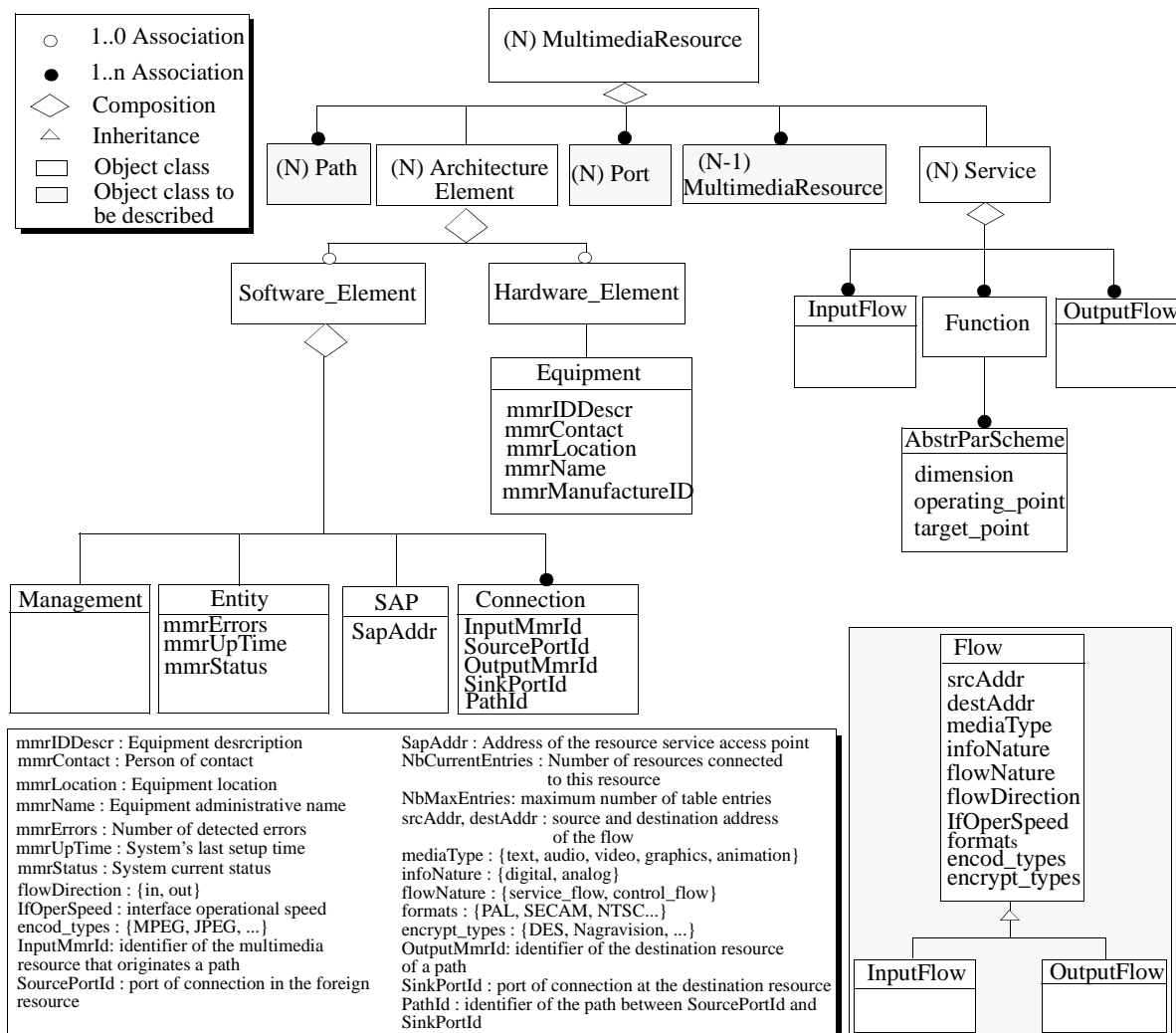


Figure 5 A generic information model of multimedia resources.

The Service class

In order to provide the service required by external entities, multimedia resources must deploy some functions for processing the incoming flows. These functions are related to the five basic management functional areas defined by ISO, which are resource *configuration*, *performance*, *security*, *fault* and *accounting* management. Some of these functions together serve to providing the QoS required from the resource. However, how can QoS be defined? According to (ITU-T E.800, 1993), the QoS is “the collective effect of service performances which determine the degree of satisfaction of a *user* of the *service*”. As such, the QoS is subjective as it relies on the perception of each service user. The given definition also restricts QoS only to the technical aspects. Accounting aspects are not properly considered. Figure 6 depicts QoS as a combination of performance, security and fault management. Obviously, a satisfactory QoS is often supported by expensive resources. Nevertheless, providing the same QoS can be cheaper from one service provider to another, depending on the adopted accounting policies. In such a paradigm, accounting derives from the provided QoS.

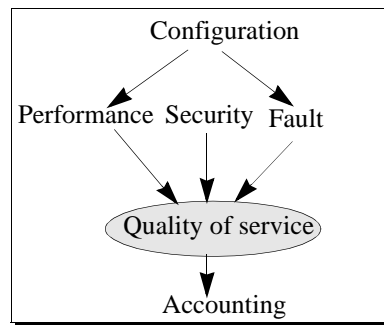


Figure 6 The basic functions involved in providing QoS.

In the opposite of accounting, configuration is implicitly present in QoS, via performance and fault management. Indeed, resource configuration influences fault and performance management. Also, detecting failures in a one-bit sampling audio CD player is easier than in a full-bit sampling audio CD player, as the former device handles fewer components.

Hence, the QoS is the main service required from a multimedia resource. The related functions (basically configuration, performance, fault and security) are characterized by some parameters. Distortion and resolution are examples of abstract parameters for the performance function. It is worth noting that those parameters are represented by their schemes. Indeed, especially in hardware-assisted devices, abstract parameters depend on some other factors. For example, the distortion parameter depends on the level of the incoming signal.

The Port class

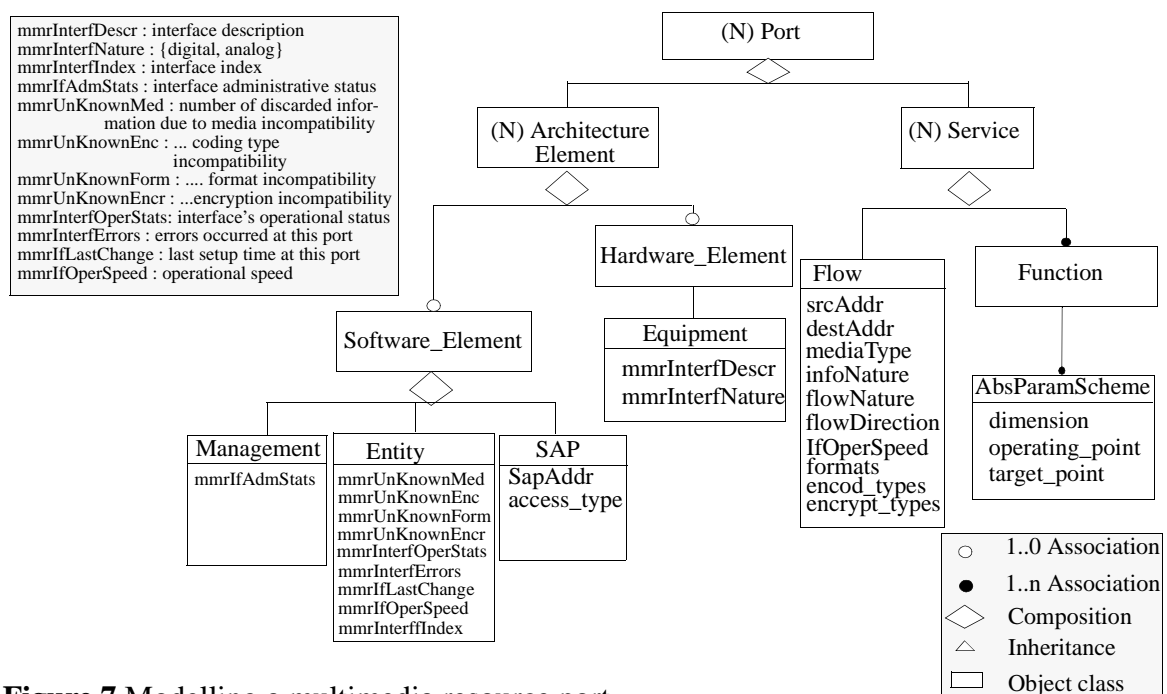


Figure 7 Modelling a multimedia resource port.

The information model of a port is depicted in Figure 7. Note that a *connection* class, used for linking two or more ports at the same level, is not present in the proposed model. The reason is

that the connection information is already present in the modelling of multimedia resources (Figure 5). Ports considered in Figure 5 are those accessible from outside the modelled multimedia resource.

The Path class

A *Path* object serves to represent the link between two *Port* objects. This link may be physical, i.e., hardware-based, or logical. An example physical path is a wire connecting two ports. In the other hand, the concept of logical path is more subtle, as the connection between the linked ports is virtual. For instance, one can specify a microprocessor for processing an information and issuing the result through a given port. In such a case, the link between the input and the output port is purely virtual. Figure 8 illustrates both mentioned aspects, namely the software and the hardware parts of a path. The software element provides a logical view of the resource as to enable the management of the underlying resource. The *Connection* objects contained in the *ConnectionGroup* object allow for expressing relationships between paths. As an example, paths that belong to the same communication session can be recorded in *Connection* objects. Therefore, synchronization issues can be accurately expressed. Moreover, this connection table (set of *Connection* objects) may serve to put priorities on the processing tasks to be held by the resource. As an example, let us assume that paths P5 and P6 (Figure 4) belong to the same communication session. The processing tasks held by MMR3 may put higher priority in treating those two paths, thereby affecting the processing of P7. Several *ConnectionGroup* objects may be needed, depending on the purposes to be achieved.

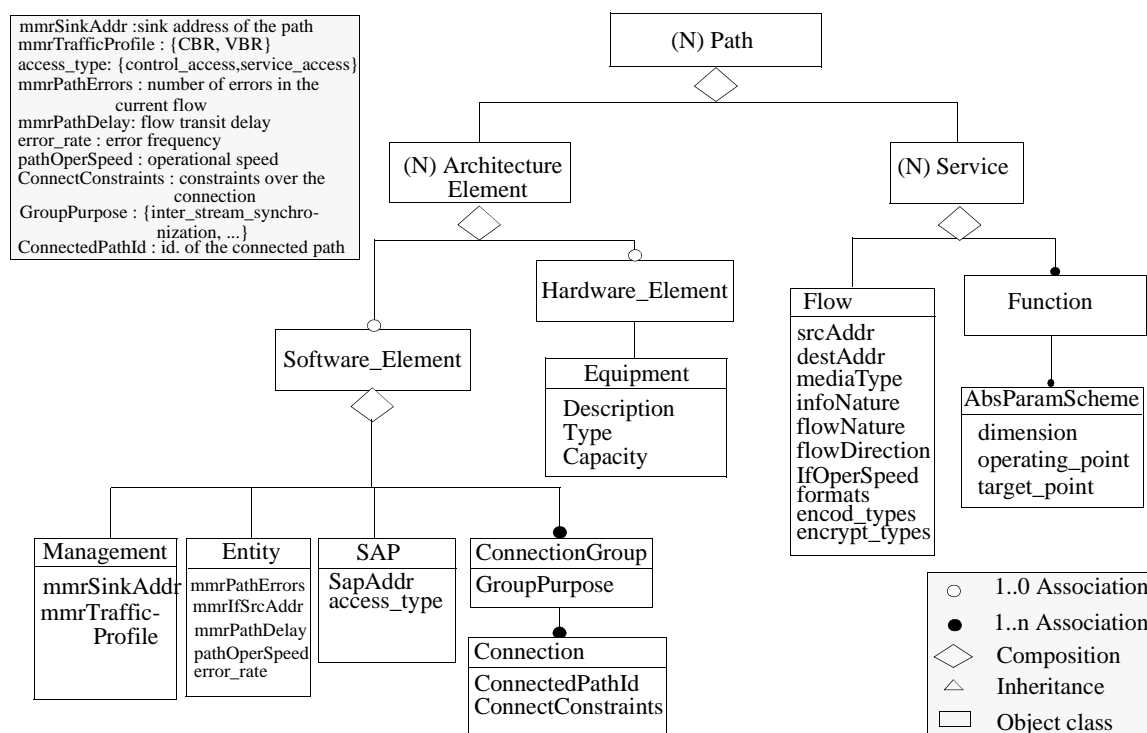


Figure 8 Modelling a Path linking two multimedia resource accesses.

In the following Section, the proposed generic information model is concretely specialized for one of the most needed multimedia resources, namely an MPEG2 video codec. The rationale for choosing MPEG2 are the service independence property and the wide range of facilities it provides.

4 CASE STUDY : AN INFORMATION MODEL FOR MANAGEMENT OF MPEG2 VIDEO CODECS

In order to facilitate the readability of the proposed information model it is worth describing first MPEG2 video codecs and their parameters that can influence the quality of a video service.

4.1 Characterization of MPEG2 video codecs

Codecs can both encode and decode information of a given coding scheme. The coding scheme considered here is MPEG2 (ISO-13818-2, 1994).

An MPEG2 codec is organized into four layers which describe its internal activities. Those are block, macroblock, slice and picture layers (Pancha and Zarki, 1994). These layers hierarchically lead to the construction of an encoded stream. Moreover MPEG2 employs both spatial and temporal compression schemes to reach high compression ratios, yet providing an acceptable perceptual video quality. The resulting outgoing frames are of three kinds: intraframes (I), interpolative frames (B) and predictive frames (P). These constitute the codec's activity modes, i.e., the codec's activities as perceived from outside. The first frame category results from spatial compression, whereas the two others stem from both spatial and temporal compression schemes. Spatial compression means reduction of the spatial redundancy. As a consequence, in sequences made up of frames resulting only from spatial compression, the loss of any of these frames does not damage the decoding of the others. On the contrary, temporal compression results in frames which may be tightly related, depending on the target compression ratio. Likewise, the resulting frames need a reference, which is an intraframe in the case of MPEG2. Hence, after coding a certain number of predictive and/or interpolative frames, a kind of reset of prediction errors is done by coding another intraframe used as the new reference and thus as the starting point of another Group Of Pictures (GOP).

The MPEG2 video compression algorithm utilizes DCT (Discrete Cosine Transform) for reducing the spatial redundancy, and block-based motion compensation for reducing the temporal redundancy. Using motion compensation serves to increase the compression ratio. Indeed very high compression ratios can be reached by using interpolative, or B, frames. These interpolate between a given frame (I or P frame) and this latter's predicted version. As a consequence, B frames introduce non-causal properties which require a re-ordering of the encoded frames before displaying the video at the reception side. Therefore, the compression ratio, yet high, leads to increasing the decoding delay. Capturing such a trade-off between codec characteristics, yet needed, is difficult to accomplish. Generic codec characteristics have been outlined in (Rodriguez, et al., 1994), as well as some influencing parameters. We have extended that work for an MPEG2 video codec (table 2). Helpful references for achieving this are (Pancha and Zarki, 1994), (Fogg, 1995) and (ISO-13818-2, 1994).

According to (Rodriguez, et al., 1994), video codecs are characterized by seven main properties: the image fidelity (i.e., how similar is the encoded picture to the original from a perceptual point of view), the bandwidth (i.e., how high is the compression ratio), the playback performance (i.e., how easy is the decoding process), the coding-to-decoding asymmetry (i.e., how similar are encoded and display aspects), the memory consumption (i.e., what amount of RAM, video buffer verifier is necessary to perform encoding/decoding activities), the processing delay, and the scalability. Scalability is the property conferred to a codec whose algorithm compresses the video in multiple frame resolutions, multiple representations of pixel color depth, and so forth, such that the encoded information is organized hierarchically with two

kinds of layers: lower or base layer, and enhancement layers. The base layer is necessary to decode the enhancement layers. Moreover, the base layer must ensure a minimum QoS.

Table 2 Capturing the main characteristics of MPEG2 video codecs

	Parameters	Influence
Image fidelity	<ul style="list-style-type: none"> - quality saturation (distortion threshold) - quantizer scale - motion estimation search range - interpolative coding - quantization matrices 	<ul style="list-style-type: none"> - variable in CBR coding to dynamically maintain the bit rate; constant in VBR and the result is a decrease of the variations in the image quality - its non-causal nature makes the video service more susceptible to errors
Bandwidth	<ul style="list-style-type: none"> - quantization scheme - traffic profile - interpolative coding (appreciated through the inter-frame to intraframe ratio N) 	<ul style="list-style-type: none"> - adaptive or predictive schemes save bandwidth - used to achieve high compression ratios
Playback performance	<ul style="list-style-type: none"> - frame rate - frame resolution - average number of color-conversion operations - quantization scheme - interpolative coding 	<ul style="list-style-type: none"> - its non-causal nature leads to out-of-sequence transmission of frames and unsuitability for real-time applications
Coding-to-decoding asymmetry	<ul style="list-style-type: none"> - pulldown - end-systems features 	<ul style="list-style-type: none"> - takes into account the difference between the source aspect ratio and the sink display aspect ratio
Memory consumption	<ul style="list-style-type: none"> - interpolative coding - picture structure 	<ul style="list-style-type: none"> - as it is non causal it leads to out-of-sequence transmission of frames - in interlaced frames, frame may be coded either as a frame or as two separately coded field pictures; these serve to save storage memory or reduce the decoding delay especially
Scalability	<ul style="list-style-type: none"> - frame resolutions - temporal resolutions - representations of pixel color depth 	
Processing delay	<ul style="list-style-type: none"> - lookup tables - quantization scheme - picture structure 	<ul style="list-style-type: none"> - simplify (color-) conversion operations - influences the number of color conversions - see above

4.2 Information model

The proposed information model (Figure 9) comprises the above characteristics as functions to be provided by the codec in order to meet a required QoS. Those functions relate either to one and/or the other tasks held by the video codec, namely encoding and decoding. Functions that mainly relate to the encoding task are image fidelity, bandwidth and scalability. The function that mainly relates to the decoding task is the playback performance. All other functions relate both to the encoding and decoding tasks. Anyway, each function is described in terms of its influencing parameters. Every such parameter aggregates some of the bitstream fields described in the normative reference (ISO-13818-2, 1994). The management of those fields is relevant to the *Mpeg2CodecManagement* instance. Therefore this latter's attributes are mainly MPEG2 bitstream fields. Note that the scope of *Mpeg2CodecManagement* is restricted to the picture layer. The systems layer, concerned, for example, with synchronization, is not considered herein. The obvious reason is that we only deal with the video codec regardless of its utilization environment.

Moreover the two coding layers we deem interesting are the frame and the slice layers. Macroblock and block layers are not considered because their introduction will add complexity to the management scheme. For the sake of clarity and conciseness, this model is not splitted in respect to the coding layers. However, the relevant parameters of those layers are considered in the global model.

The functions mentioned in Figure 9 are deployed in order to process the two kinds of flows treated by an MPEG2 video codec, namely *PictureFlow* and *Bitstream*. The former relates to the pictures "captured" by the video codec for encoding purpose, and the latter describes the encoded bitstream resulting from the encoding process. The decoding stage is very symmetric, i.e., the bitstream becomes the incoming flow while the picture flow is the outgoing flow. The notation used for designing the model does not permit expression of such a relationship as that stating the dual behaviour of a bitstream which can be either an outgoing or an incoming flow. Anyway, every bitstream is composed of frames which are base layer encoded, or enhancement layer encoded. Each such layer derives from a generic *QoSLayer*. Moreover the various types of scalability are not considered separately. Instead their meaningful parameters are set as attributes of the layer objects, hence allowing hybrid enhancements.

With respect to the *Software_Element* object, it also aggregates an *MpCodecEntity* object in addition to the *Mpeg2CodecManagement* object. The former object stores some information about the codec performance such as its bit error rate (BER), the number of broken links (number of B-pictures undecoded because of corruption of the reference frame), the number of unrecoverable frame errors and the number of recovered frame errors. That means likely provision of recovery mechanisms to increase the playback performance.

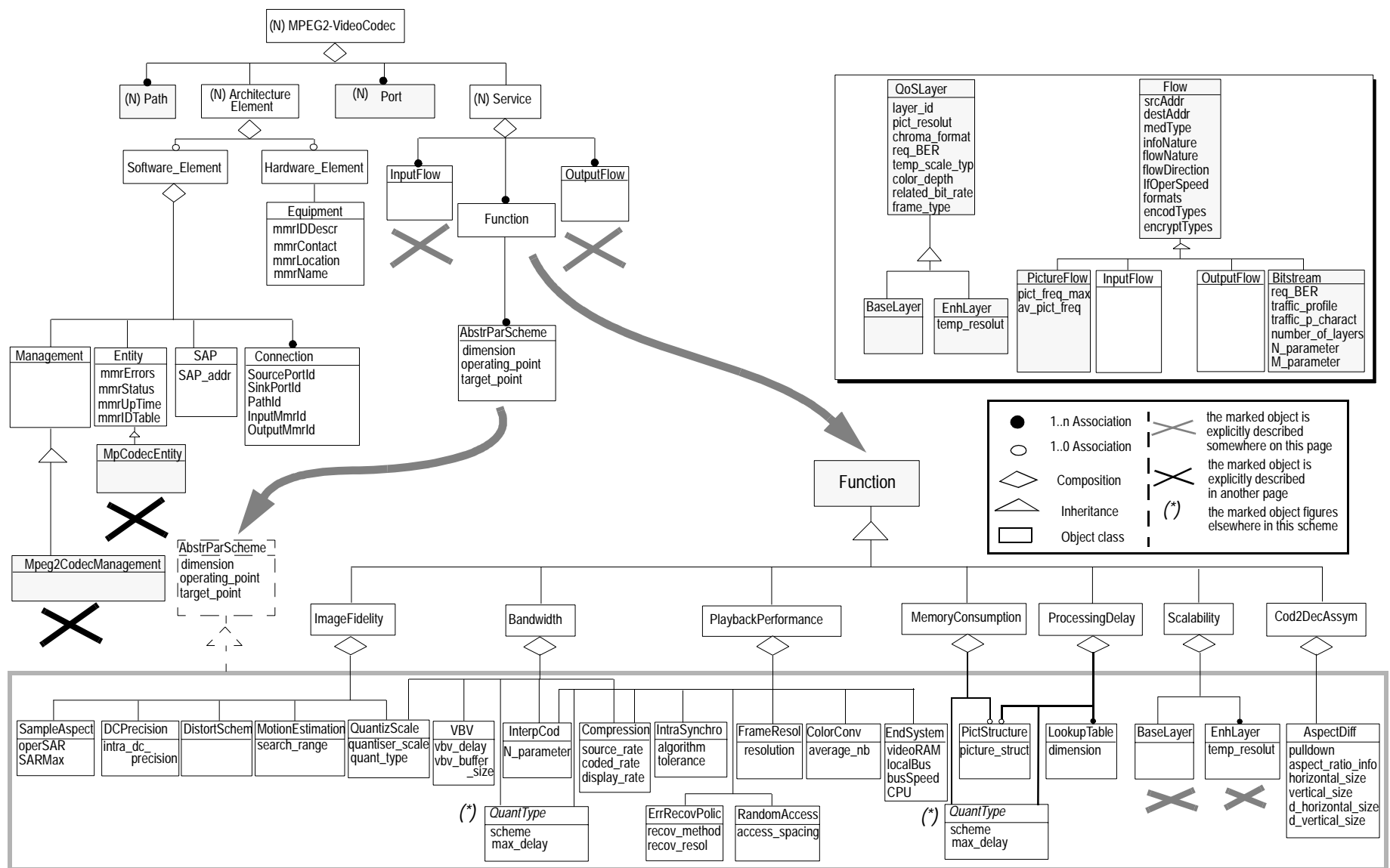


Figure 9 Modelling an MPEG2 video codec.

Mpeg2CodecManagement	MpCodecEntity
<ul style="list-style-type: none"> TestModel N_parameter M_parameter Error_Recovery chroma_format progressive_sequence picture_structure dct_type progressive_frame aspect_ratio_information horizontal_size vertical_size frame_rate_code user_data video_format matrix_coefficients display_horizontal_size display_vertical_size scalable_mode intra_dc_precision vbv_buffer_size bit_rate horizontal_subsampling_factor_m horizontal_subsampling_factor_n vertical_subsampling_factor_m vertical_subsampling_factor_n pict_mux_enable 	<ul style="list-style-type: none"> mux_to_progressive_sequence closed_gop frame_pred_frame_dct composite_display_flag field_sequence intra_quantiser_matrix non_intra_quantiser_matrix chroma_intra_quantiser_matrix chroma_non_intra_quantiser_matrix frame_centre_horizontal_offset frame_centre_vertical_offset forward_temporal_reference backward_temporal_reference lower_layer_temporal_reference lower_layer_horizontal_offset lower_layer_vertical_offset spatial_temporal_weight_code_table_index lower_layer_progressive_frame lower_layer_deinterlaced_field_select quantiser_scale_code q_scale_type number_of_layers time_code f_code[s][t] output_format constrained_parameters slices_per_frame
<div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>operSAR : operational Sample Aspect Ratio</p> <p>SARMax : max. value of SAR</p> <p>intra_dc_precision : precision of DC coefficients</p> <p>search_range : search range for estimating the motion vector</p> <p>quantiser_scale : quantization scale</p> <p>quant_type : quantization type</p> <p>VBV : Video Buffer Verifier</p> <p>vbv_delay : time needed to fill the VBV buffer</p> <p>vbv_buffer_size : size of VBV needed to decode the sequence</p> <p>N_parameter : interframe to intraframe ratio</p> <p>M_parameter : intraframe to predictive frame ratio</p> <p>scheme : quantization scheme</p> <p>max_delay : maximum delay introduced by the quantization</p> <p>source_rate : rate of pictures to be encoded</p> <p>coded_rate : bit rate of encoded frames</p> <p>display_rate : rate of displayed pictures</p> <p>IntraSynchro : intra-medium synchronization</p> <p>algorithm : intra-medium synchronization method</p> <p>tolerance : intra-medium synchronization tolerance</p> <p>ErrRecovPolic : Error recovery policies</p> <p>recov_method : recovery method</p> <p>recov_resolution : recovery resolution, e.g., the number of consecutive corrupted frames the method can help recover from</p> <p>RandomAccess : random access to fields of the bitstream</p> <p>access_spacing : minimum spacing between fields to access</p> <p>FrameResol : Frame resolution</p> </div> <div style="width: 48%;"> <p>localBus : type of local bus used by the end-system, if any.</p> <p>busSpeed : efficiency of the end-system's bus</p> <p>CPU : Central Processing Unit</p> <p>picture_struct : picture structure ::= { top_field, bottom_field, frame_picture}</p> <p>temp_resolut : temporal resolution</p> <p>pulldown : presentation aspect difference between the encoded pictures and the original ones</p> <p>d_horizontal_size : display horizontal size</p> <p>d_vertical_size : display vertical size</p> <p>AspectDiff : aspect difference</p> <p>layer_id : layer identifier</p> <p>pict_resolut : picture resolution</p> <p>req_BER : required Bit Error Rate</p> <p>temp_scale_type : type of temporal scalability</p> <p>color_depth : number of colors by pixel</p> <p>related_bit_rate : bitrate related to the QoS layer</p> <p>frame_type : field frame or picture frame</p> <p>pict_freq_max : max. value of the picture frequency</p> <p>av_pict_freq : average picture frequency</p> <p>traffic_profile : traffic profile ::= (Constant_Bit_Rate, Variable_Bit_Rate)</p> <p>traffic_p_charact : characteristics of the used traffic profile</p> <p>number_of_layers : number of layers, including the base layer</p> <p>aspect_ratio_info : SAR of reconstructed frames</p> <p>ColorConv : color conversion</p> <p>average_nb : average number of color conversion operations</p> <p>videoRAM : video RAM provided by the end-system</p> </div> </div> <p style="text-align: center;">Non mentioned parameters are referenced in (Fogg, 1995)</p>	

Figure 9 (cont'd)

4.3 Example use of the model

The designed information model can be used for managing a multimedia service as depicted in Figure 10. In this example, the scene captured by the camera is first MPEG2-encoded and then transmitted over the network using a network adapter. At the reception side, the transmitted streams are decoded and displayed. The service management architectural unit is responsible for setting accurately the relevant parameters of the resources (codecs, network resources, etc.) involved in the session. That unit uses the codecs' information model to know about the information to be managed, and then performs the mechanisms needed to cater for the users QoS requirements. For instance, how far could a network dysfunction be compensated by a better efficiency of video codecs? Therefore, the design of a management framework, thereby the underlying information model, is definitely justified. Indeed, the information model proposed

for MPEG2 video codecs features the parameters to be tuned in order to meet the users requirements. Mechanisms to be invoked for this tuning as well as the estimation of the accurate parameter values to be set are implementation and experiment concerns. Hence, three worlds are now converging, which are network management, signal processing and computing, in order to build up a consistent and integrated service management scheme.

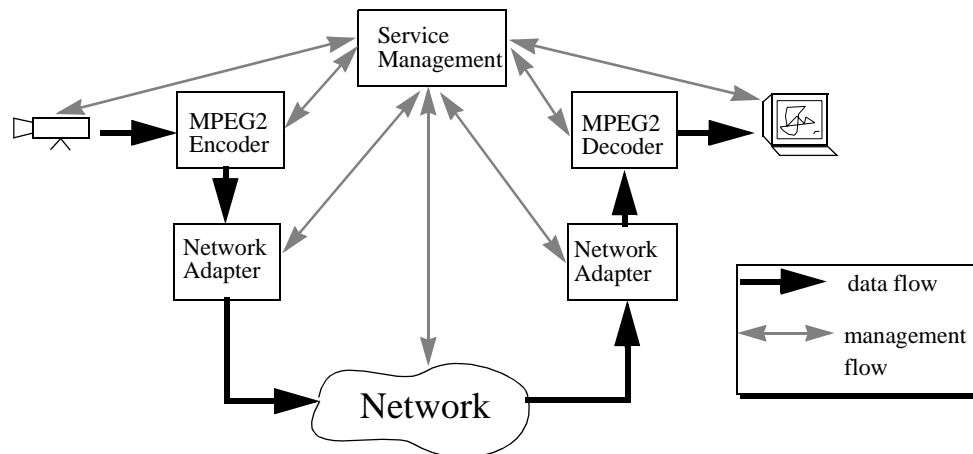


Figure 10 Example use scheme of the designed MPEG2 codec information model.

5 CONCLUSION

The main achievements of this document are the design of a generic information model of multimedia resources, and the specialization of this model for a concrete resource, namely an MPEG2 video codec, on the basis of the related standards draft (ISO-13818-2, 1994). To improve the designed information model, an implementation is needed. The goal thereof will be to explore how the parameters encapsulated as attributes in the *Mpeg2CodecManagement* entity influence the codec activities, in terms of the provided QoS. Some other parameters may be needed to fully exploit the possibilities offered by the MPEG2 coding scheme. Anyway the present paper is a very first step towards an accurate modelling of multimedia resources in general, and MPEG2 video codecs in particular. This goal will depend on the advances in the signal processing area, with respect to the parameters which are crucial for the efficiency of MPEG2 codecs.

This work is part of a wider project called OAMS (Open management Architecture for Multimedia Services over ATM) (Znaty, et al, 1995). The OAMS project aims at designing and developing a management architecture for the management of advanced telecommunication services or multimedia services and the management of the underlying networks that are used to provide those services. In this context, the work presented in this paper is a contribution to service management.

6 ACKNOWLEDGEMENTS

We would like to acknowledge Giovanni PACIFICI, Rolf STADLER and Mun Choon CHAN from the Center for Telecommunications Research (CTR), Columbia University, NY, for providing reference (Chan, et al., 1995). Likewise, special thanks go to Xavier LOGEAN, Slim GARA, Frédéric TORASSO and Andrea BASSO for their review, and pertinent remarks and advice.

7 REFERENCES

- (Chan, et al., 1995) Chan, M.C., Pacifici, G., Stadler, R. (May 1995). *Objective-Driven Performance Management for Broadband Networks*. CTR Technical Report 404-95-10, Columbia University, NY.
- (Fogg, 1995) Fogg, C. (1995). *MPEG FAQ 4.0*. Available at <http://www.cs.tu-berlin.de/~phade/mpegfaq/index.html>.
- (Gbaguidi, 1995) Gbaguidi, C. (1995). *Towards Definition of a Management Information Base for Multimedia Resources*. Working document.
- (ISO-13818-2, 1994) Draft International Standard ISO/IEC/ DIS 13818-2. (1994). *Information Technology- Generic Coding of Moving and Associated Audio Information- Part2 : Video*.
- (ITU-T E.800, 1993) ITU -T E.800. (1993). *Quality of Service and Dependability Vocabulary*.
- (ITU-T X.711, 1991) ITU-T X.711 (1991). *Common Management Information Protocol Specification for CCITT Applications*.
- (Lazar, et al, 1995) Lazar, A. A., Ngoh, L.H, Sahai, A. (1995). *Multimedia Networking Abstractions with Quality of Service Guarantees*. Available at <ftp.ctr.columbia.edu/CTR-Research/comet/public/papers>.
- (Pancha and Zarki, 1994) Pancha, P., El Zarki, M. (May 1994). *MPEG Coding For Variable Bit Rate Video Transmission*. IEEE Communications.
- (RFC1155, 1990) Rose, M., McCloghrie, K. (May 1990). *Structure and Identification of Management Information for TCP-IP based internets*. RFC 1155, IETF.
- (RFC1157, 1990) Case, J., Fedor, M., Schoffstall, M., Davin, J. (May 1990). *Simple Network Management Protocol*. RFC 1157, IETF.
- (Rodriguez, et al., 1994) Rodriguez, A. A., Morse, K. (Fall 1994). *Evaluating Video Codecs*. IEEE Multimedia.
- (Sclavos, et al., 1994) Sclavos, J., Simoni, N., Znaty, S. (1994). *Information Model: From Abstraction to Application*. IEEE NOMS, Florida, USA.
- (Voruganti, 1994) Voruganti, R. R. (August 1994). *A Global Network Management Framework for the '90s*. IEEE Communications Magazine.
- (Znaty, et al, 1995) Znaty, S., Gbaguidi, C., Hubaux, J.-P. (1995). *OAMS : An Open Management Architecture for Multimedia Services over ATM*. Internal Report, 1995.